CHAPTER

4

DOMAIN NAME SYSTEM (DNS)

After completing this chapter, you will be able to:

- Understand DNS name resolution
- Understand the different types of name servers
- Configure and manage DNS zones
- Understand zone transfers
- Install DNS for Active Directory
- Configure DNS for Active Directory
- Monitor and troubleshoot DNS for Active Directory

The Domain Name System (DNS) is a name resolution database that is used on Transmission Control Protocol/Internet Protocol (TCP/IP) networks and is associated primarily with the Internet. Everyone who has used the Internet is familiar with DNS names—for example, www.InsideIS.com or ftp.Microsoft.com—whether they realize it or not. However, DNS also plays an integral role in an Active Directory implementation. Essentially, Active Directory cannot function without DNS (either an implementation of Microsoft DNS or third-party DNS server software).

In this chapter, you will learn not only what DNS is and how to set up, configure, monitor, and troubleshoot it, but also about the advantages and disadvantages of using Windows 2000 DNS. By the end of this chapter, you should be able to deploy DNS on your network for use with Active Directory.

Understanding DNS Name Resolution

The Domain Name System originated from the Internet, which is based on the TCP/IP network protocol. Computers communicate with each other through numbers, and on TCP/IP networks that means IP addresses. An IP address is formed of four octets of numerals from 1 to 254. For example, **192.168.1.21** is a valid IP address. Part of the address refers to a network ID signifying the network the computer is on, and the remaining part is the host ID, which uniquely identifies a given host on a TCP/IP network. A process called **masking** determines where the network ID stops and the host ID begins, which is why you often see the term **subnet mask** used in conjunction with IP addresses.

HOSTS Files and Their Function

Unfortunately, humans are not as good at remembering numbers as computers are. In fact, we work much better with names. The process of **name resolution** was created to enable humans to work more efficiently with networked computers. Originally, all the host-to-IP address mappings were kept in a single file called HOSTS. Whenever a new computer was introduced to the Internet, a systems administrator would manually update the HOSTS file and send copies of it to all the other systems on the Internet.

Although this was an adequate solution in the early days of the Internet, when there were few systems and updates were infrequent, the manual process quickly reached its limit and became a burden. In addition, the HOSTS file provided only what is called a **flat** namespace. Because the namespace was flat, any change made to a rapidly growing HOSTS file—such as adding a single computer—had to be replicated to every system on the Internet. Every system was responsible for being able to resolve the name of every host to its IP address. To alleviate the growing problem of maintaining an up-to-date HOSTS file, DNS was born.

Name Resolution Beyond HOSTS: DNS

DNS created a **hierarchical** namespace, which allowed the namespace of the Internet to be partitioned and distributed. This arrangement eliminated the need to distribute the entire database to every host. Name servers could be set up that were responsible for only a portion of the namespace rather than all of it. For example, the DNS server ns1.InsideIS.com has to be responsible only for the InsideIS.com namespace, not the entire Internet. If a host on the InsideIS.com network needs to resolve the name of a host outside that domain, the name server knows how to contact other name servers that are responsible for different pieces of the overall domain namespace.

At the top of the namespace is the root, which is represented by a period (**.**). Below the root domain are the top-level domains. Table 4-1 illustrates the more popular top-level domains and the type of organizations with which they are associated.

Table 4-1 Common top-level domains

Top-Level Domain	Type
Com	Commercial organization
Edu	Educational institution
Gov	U.S. government department
Mil	U.S. military organization
Net	Network provider
Org	Nonprofit organization

Although the types listed are the traditional designations, the main domain name registrar—Network Solutions—allows just about anyone to register a Com, Net, or Org domain name without regard to how it will be used. Now you'll see commercial companies with **.org** names, and home users/consumers with registered **.net** names, simply because the **.com** equivalent is already taken. At one time, however, these designations were fairly strictly enforced, and to register a name you had to show the intended purpose of the site.

Second-Level Domains

Below the top-level domains are the second-level domains, which are the domain names that individuals and organizations register with a registrar such as Network Solutions. These are names like Inside-Corner.com, Examcram.com, Whitehouse.gov, and Army.mil. The domain namespace can be partitioned even further. InsideIS.com can yield www.InsideIS.com, ftp.InsideIS.com, and so forth. Figure 4-1 shows an example of the domain namespace.

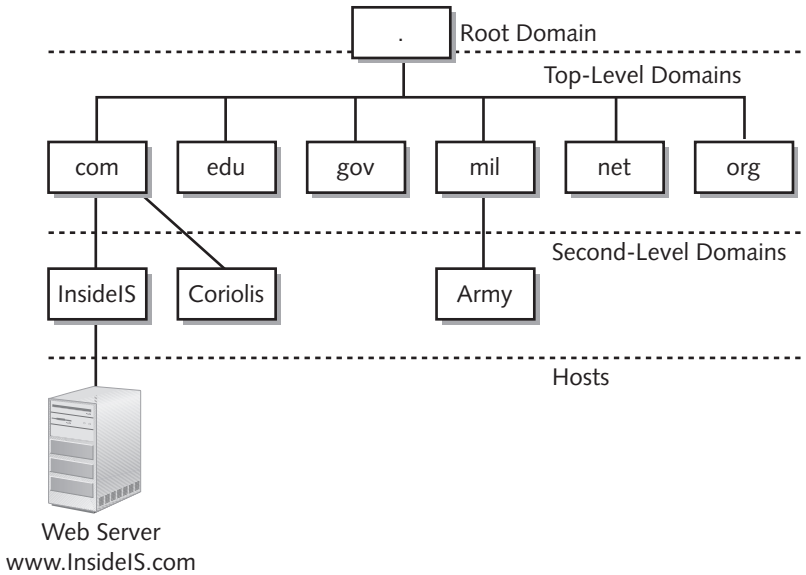

Figure 4-1 An example of the domain namespace

Fully Qualified Domain Names

In our previous example, the host computer www.InsideIS.com. is what is known as a fully qualified domain name (FQDN). Notice that the trailing period after .com is intentional. A FQDN describes the exact position of a host within the namespace. In this case, **www** is the computer, **InsideIS** is the second-level domain of which **www** is a member, **.com** is the top-level domain, and the trailing **.** represents the root domain. The DNS system uses a host's FQDN to resolve its name to an IP address.

Relative Distinguished Names

In situations where hosts are on the same network, using an FQDN to communicate between them is cumbersome. To compensate, DNS provides for a **relative distinguished name**, which is simply the host name. In our www.InsideIS.com FQDN example, **www** is the relative distinguished name. This name provides a more convenient means of communicating with other systems that belong to the InsideIS.com second-level domain, reducing redundancy.

Active Directory and DDNS

Windows 2000 is built to run on TCP/IP, and to utilize Active Directory you must forsake the older Windows Internet Naming System (WINS) technology for DNS. Probably the biggest downside to DNS is that, while distributed, it was still designed as a system that required manual updates. Whenever a new host was added to a domain, an administrator needed to update manually the zone database on the primary DNS server (zones are discussed later in this chapter) to contain the new host. If the network included secondary name servers, the changes were replicated in a zone transfer.

Recently, a dynamic updating feature was proposed in RFC 2136 that provides the means for updating a zone's primary server automatically. Windows 2000 supports this new Dynamic DNS (DDNS). The caveat is that it works only with Windows 2000 clients—older Windows NT, Windows 9*x*, and non-Windows clients must still be added manually. When DDNS is enabled and a Windows 2000 client logs into the network, the client automatically sends an update to the name server it has been configured to use, adding its A (address) record. This process greatly simplifies the administration of DNS on a Windows 2000 network.

Now that you have a basic understanding of what DNS is, let's look more closely at exactly how the name resolution process works.

Forward Lookup Queries

As we've said, part of the beauty of the DNS system is that the database can be distributed so that not every name server has to be responsible for resolving every host on the network. So, unlike the days when TCP/IP networks relied strictly on HOSTS files, no one server knows how to resolve every host name to an IP address on the Internet. A

DNS server could be all-knowing for a given company, but, unless that company does not communicate with any outside networks, eventually any DNS server will need to ask for help to resolve a name.

The standard method of name resolution with DNS is the **forward lookup query**. In this method, the client needing to resolve a host name to its IP address sends a query to its primary DNS server (configured on the client) asking for help. The DNS server checks its zone database to determine if it contains the host-to-IP address mapping to answer the client query. If so, it returns the information to the client to fulfill the query. If not, it forwards the request to another DNS server for resolution. This process continues until the FQDN is completely resolved; then the information is passed back the way it came.

For example, suppose we attempted to access www.examcram.com from our current computer, a host in the inside-corner.com domain:

1. When we enter "www.Examcram.com" into our browser, our system contacts our primary DNS server as configured on our system.
2. The local DNS server is authoritative only for the inside-corner.com domain, so it forwards the query to a root server on the Internet.
3. The root server responds by informing our primary DNS server that www.examcram.com is in the Com domain, and that it needs to talk to a Com DNS server.
4. Our DNS server contacts a Com DNS server with the query; this server in turn responds with a referral to the name server authoritative for the ExamCram.com domain.
5. Our DNS server contacts the name server authoritative for ExamCram.com, which finds the correct host-to-IP address mapping and returns the results of the query to our DNS server, which then returns the results to our host computer.
6. Communication can be established directly between our computer and www.examcram.com. In this case, the Web site would load into our browser.

Caching

As you can probably imagine, the name resolution process can result in a lot of network traffic, because a query can potentially be sent to numerous name servers in order to be resolved. To alleviate this problem, DNS servers use a process called **caching**: The DNS server will store the information it learns about host-to-IP address mappings outside of its zone for a configurable length of time. This way, client queries for the same host in the near future do not have to repeat the entire resolution process.

The configurable length of time a DNS server will cache the results of a query is referred to as Time To Live (TTL); in Windows 2000, it defaults to 60 minutes. Longer TTL values reduce network traffic for subsequent communication with remote hosts, whereas

shorter TTL values ensure that information about the namespace is current. This happens because if a remote host's DNS information changes while information about it is cached, a client won't know about the changes until the TTL expires and the name server initiates a new name resolution query.

Reverse Lookup Query

In some cases (most notably, troubleshooting) you need to be able to query in reverse—that is, to resolve a name from an IP address. The regular forward lookup query process that DNS uses is much like a phone book: You look up someone's name and find the phone number. Typically, you don't start with a phone number and want to find the name associated with it. However, some applications implement security based on being able to resolve a name from an IP address, and troubleshooting tools like the **nslookup** command use **reverse lookup queries** to report host names from IP addresses.

Because the DNS namespace is distributed as host-to-IP address (like a phone book), a reverse lookup query would require exhaustively searching every domain name until the correct host was found. To get around this problem, a special second-level domain was created that contains only IP address-to-host mappings. This second-level domain is called in-addr.arpa, and it follows the same hierarchical structure as the rest of the DNS namespace. Subdomains in the in-addr.arpa domain are named in reverse dotted-decimal format. For example, the network ID **192.168.1.0** would be represented as **1.168.192.in-addr.arpa**. Troubleshooting with nslookup is covered later in this chapter, when we'll show you reverse lookup queries in action.

Understanding the Different Types of Name Servers

In our previous discussion, we've really considered only one type of DNS server: a primary DNS server. However, there are other types of name servers. We'll cover the following types in this section:

- Primary DNS servers
- Secondary DNS servers
- Caching-only name servers
- Forwarding DNS servers

Primary DNS Servers

There are two main types of DNS servers: primary servers and secondary servers. As we briefly mentioned earlier, the DNS namespace is partitioned into what are known as **zones**. We discuss zones in detail later in this chapter, but for now you should just understand that a zone is the part of the overall DNS namespace that is controlled by a primary server. There can be only one primary server in a zone, and that primary zone is said to be **authoritative** for the zone. It is the master, and any changes to the DNS

domain must be made on the primary server. The exception is an Active Directory–integrated zone, which is also discussed later in this chapter.

Secondary DNS Servers

A secondary server is basically a backup server for the primary server. It is important to note that during the name resolution process, if a primary server cannot resolve a host name, the query is *not* submitted to a secondary server (if one exists in the zone). Essentially, the secondary server is used as a failover, which means it does essentially nothing until the primary server fails, at which time the dormant secondary server picks up where the failed server left off with no interruption in service to the user. If a client is unable to contact the primary DNS server, it will attempt to use the secondary server, if one has been configured. Another potential use of a secondary server is load balancing. If you have 1,000 network clients, for example, you could configure half of them to use the primary server first and half of them to use the secondary server first. This setup would reduce the load on the primary server.

Changes are never made directly to a secondary server, which receives a copy of the master zone file from the primary name server in a zone. This process is called **zone transfer** and is covered in more detail later in this chapter. Unlike with primary servers, a zone can have multiple secondary servers.

Caching-Only Name Servers

A caching-only name server does pretty much what the name implies: It functions only to cache queries. The caching-only name server does not maintain a zone database file, nor does it receive updates from a primary server. It simply performs queries, caches the results, and returns results to querying clients.

The advantage of using a caching-only name server is a twofold reduction in network traffic. First, no replication traffic is generated between the primary name server and the caching-only server, as happens between a primary and secondary. Second, a caching-only server reduces name resolution traffic by reducing the need for subsequent queries to go through the entire name resolution process.

Caching-only servers are still bound by TTL rules, although the TTLs are often set longer than they are on primary and secondary servers. TTLs are set longer on caching-only servers since the goal of having this type of DNS server is to build up a substantial cache to reduce name-resolution traffic. The disadvantage of caching-only servers is that, if a server is rebooted, the cache is flushed and the server must build its cache again from scratch.

Caching-only servers can also perform what is called **negative caching**, which caches failed results. This reduces the timeout process when a client queries for a site that does not exist or is unavailable.

Forwarding DNS Servers

Forwarding DNS servers exist solely to communicate with DNS servers outside the local zone. By default, any DNS server that receives a query it cannot resolve will contact an outside DNS server in order to resolve the name for the client making the query. A DNS forwarder functions like a proxy, becoming the only DNS server in a zone that can communicate outside the zone. For example, if the primary name server cannot resolve a name, it will send the query to the forwarding DNS server for resolution. Figures 4-2 and 4-3 show a DNS infrastructure not using a forwarder and one using a forwarder, respectively.

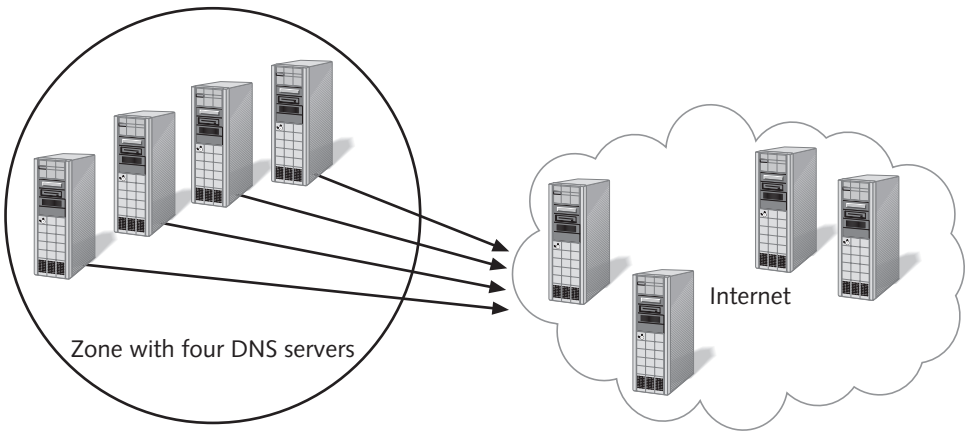

Figure 4-2 A DNS zone in which all name servers communicate outside the local zone

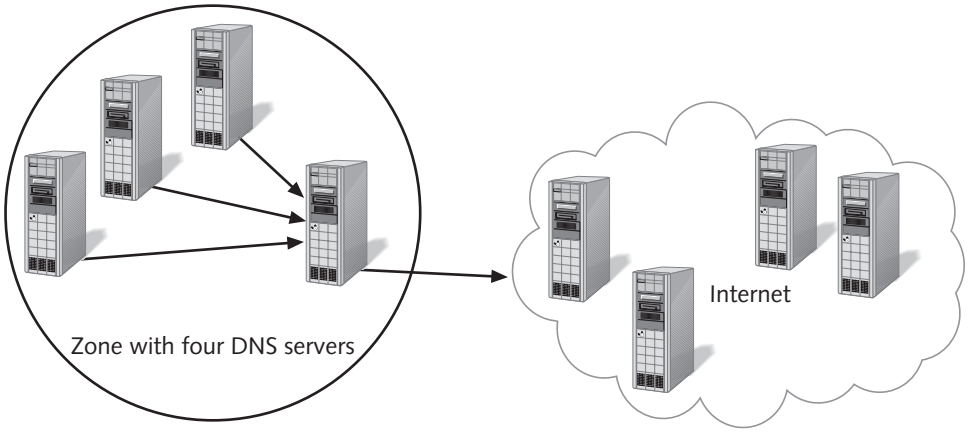

Figure 4-3 A DNS zone that uses a forwarding name server to communicate outside the local zone

Forwarding servers can be configured to use either nonexclusive or exclusive mode. In nonexclusive mode, a name server can attempt to resolve a query through its own zone database files if a forwarder cannot resolve the query. In exclusive mode, if a forwarder cannot resolve a query, the server that sent the query to the forwarder does not attempt to resolve the name itself, and simply returns a failure notice to the client that originated the request.

Configuring and Managing DNS Zones

We've mentioned zones a few times so far, but we haven't taken the time to talk about them in much depth. As we've said, a zone is a partitioned portion of the overall DNS namespace. Zones make managing the namespace much easier than the flat namespace of HOSTS files did. Figure 4-4 shows an example of the InsideIS.com domain divided into two zones.

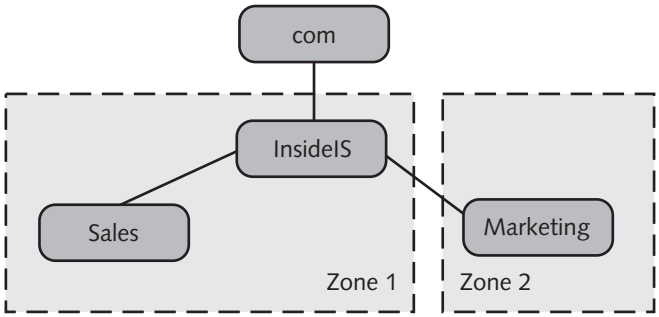

Figure 4-4 An example of DNS zones

At the simplest level, a single zone contains the entire namespace of a second-level domain. In our example, a single zone could be authoritative for all of InsideIS.com without dividing it into two zones. A zone must encompass contiguous namespace, however, which means a single zone could not be authoritative for both InsideIS.com and ExamCram.com. Those two domains are not part of the same namespace.

In our example, we divided our namespace into two zones, in which one zone is authoritative for Sales.InsideIS.com and a second zone is authoritative for Marketing.InsideIS.com. You will use multiple zones primarily to distribute administrative responsibilities. In many corporations, political boundaries must be managed, with different divisions/departments having their own administrators. Multiple zones allow multiple administrators to be responsible for their individual pieces of the namespace.

Another reason to partition the namespace into zones is to reduce the load on a DNS infrastructure. Consider a megacorporation such as Microsoft, with more than 100,000 nodes on the network, spread across the globe. A single zone would place a tremendous burden on the primary DNS server (remember, a zone can contain only one primary

server), and the replication traffic to secondary DNS servers would make a significant impact on network performance. Dividing the Microsoft.com namespace into multiple zones distributes the load, which increases performance and eases administration.

Windows 2000 Zones

Windows 2000 supports two types of zones: forward lookup zones and reverse lookup zones. These zones are associated with the types of name resolution queries they enable. We will discuss these zones in greater detail later in this chapter when we look at installing and configuring Windows 2000 DNS.

Understanding Zone Transfers

Zone transfer is the process by which changes made on the primary server are replicated to all of the secondary servers in the zone. There are three types of zone transfers to consider:

- Full zone transfer
- Incremental transfer
- DNS Notify

Full Zone Transfer

Originally, the only method of replication between primary and secondary servers was the full zone transfer. With this method, the entire zone database file is transferred whenever an update is made. Zone transfer is performed through a "pull" mechanism rather than a "push"—that is, the secondary servers initiate a zone transfer. The process is as follows:

1. The secondary server waits a predetermined amount of time before contacting the primary server. When it does establish contact, it requests the primary server's SOA (Start of Authority) record. Record types are discussed in depth below.
2. The primary server responds to the secondary server with its SOA record.
3. Whenever a change is made to the primary name server, the serial number held in the SOA record is incremented. When the secondary server receives the SOA record from the primary server, it compares the serial number to its own. If the serial number in the SOA record sent by the primary server is higher than the serial number in the SOA record currently on the secondary server, the secondary server knows its zone database is out of date. It then sends a request back to the primary server for a full zone transfer. This full transfer is done through an AXFR (or Full Zone Transfer) request.
4. The primary server sends its full zone database file back to the secondary server. After the update is complete, the process begins again with the waiting period.

Incremental Transfer

As you can probably imagine, performing a full zone transfer every time a change is made to the primary server is inefficient. It also can generate a lot of network traffic if the primary server receives frequent updates and there are multiple secondary servers. To get around this problem, RFC 1995 allowed for incremental zone transfers. As the name implies, with an incremental transfer only the portion of the database that has been changed is replicated.

The process with an incremental transfer is basically the same as a full zone transfer. The difference lies in the type of request. During an incremental transfer, the secondary server sends an IXFR (or Incremental Zone Transfer) request signifying an incremental transfer, rather than an AXFR request signifying a full zone transfer.

Version History

So, how do the name servers keep track of the changes in order for incremental transfer to work? The primary server maintains a version history, which keeps track of all changes that have been made since the last version update was transferred to a secondary server. When a secondary server requests an IXFR transfer, the primary server begins sending the recent updates, starting with the oldest updates and progressing to the most recent updates.

When the secondary server begins receiving the updates, it creates a new version of the zone and begins applying the updates to that copy. After all the updates are committed to the copy of the zone database, the original database is replaced with the copy.

If the primary server does not support incremental transfers, it will simply ignore the incremental request of the secondary server and perform a full zone transfer. A primary server that supports incremental transfers can also arbitrarily decide to perform a full zone transfer even if an incremental transfer is requested, if conditions indicate that a full zone transfer would be better. For example, if a large number of updates have been made since the last transfer, an incremental transfer would consume more bandwidth than a full zone transfer.

DNS Notify

DNS Notify was proposed in RFC 1996 as an update to incremental transfer. With DNS Notify, rather than waiting for a secondary server to contact the primary server to see if there are any changes, the primary server notifies secondary servers in its notify list whenever an update is made. The notify list is maintained on the primary server; it contains a list of IP addresses of secondary servers that should be notified whenever an update is made. This process helps the zone database stay more consistent across the enterprise. The DNS Notify process works as follows:

1. When the zone is updated on the primary server, the serial number in the SOA record is updated to reflect a newer version of the zone.

2. The primary server consults its notify list and contacts members of that list, informing them of a newer zone database version.
3. Similar to the process with a full zone transfer or an incremental transfer, the secondary server contacts the primary server and requests the SOA record.
4. When the SOA record is received, the secondary server compares the serial numbers of the primary server's SOA record and its own SOA record.
5. If the serial number of the primary server is higher, the secondary server knows its version of the database is out of date. It then requests a zone transfer (AXFR or IXFR).

Installing DNS for Active Directory

So far, we've discussed a lot of DNS theory, but not much of the practical side of things. In this section, we'll cover actually installing and configuring the DNS service for Windows 2000.

DNS Preinstallation

We must consider a few preinstallation tasks before starting the installation procedure. The first of these tasks is making sure your server is ready for DNS. A DNS server, for obvious reasons, must have a static IP address. If the server that is to run the DNS service currently has a Dynamic Host Configuration Protocol (DHCP)–assigned IP address, you must reconfigure that before continuing. To configure the IP address in Windows 2000, perform the following steps:

1. Right-click on My Network Places and click on Properties. A window similar to that in Figure 4-5 will appear.
2. Right-click on the network adapter that the DNS service will use and click on Properties. A window similar to that in Figure 4-6 will appear.

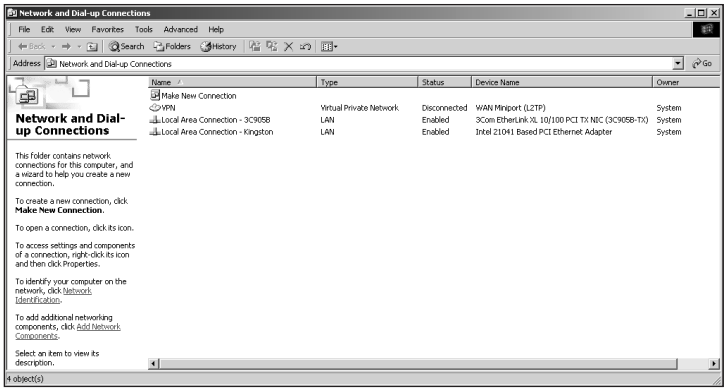

Figure 4-5 You can configure network settings through Network and Dial-up Connections Properties

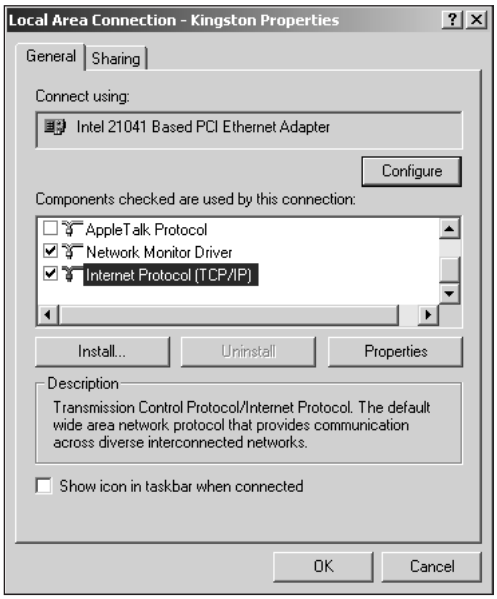

Figure 4-6 You can configure settings for your network adapter through its property page

3. Navigate down the Components list until you see Internet Protocol (TCP/IP). Right-click on TCP/IP and click on Properties to open the property sheet where you can configure the IP address, or left-click to highlight and then choose the Properties button at the bottom of the screen. This window is shown in Figure 4-7.

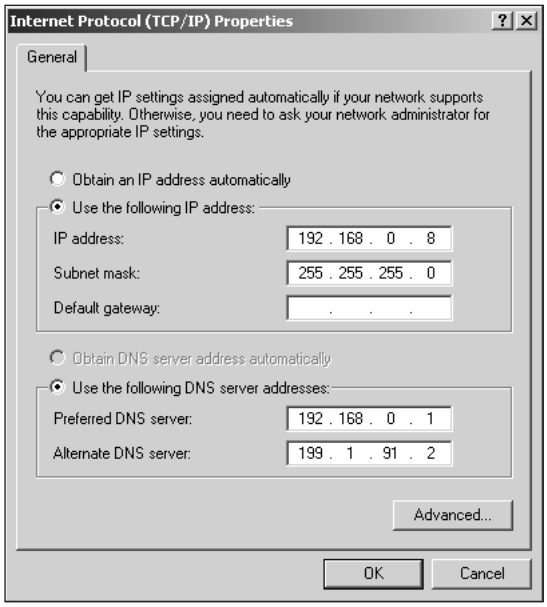

Figure 4-7 In TCP/IP Properties, you can configure TCP/IP settings for your network adapter

4. In our example, our internal network is using the standard reserved class C network ID. You should configure your server to use an IP address that is valid for your subnet.
5. Before exiting this dialog box, you must perform an additional step: configuring the DNS server IP address under the Advanced TCP/IP Settings. Click on the Advanced button, and then click on the DNS tab. You will see a window similar to that shown in Figure 4-8.

In our example, the first listed DNS server is primary and the second server is secondary.

DNS Installation

For the sake of this discussion, we will assume that you did *not* choose to install the DNS service when you initially set up Windows 2000. It is not installed by default, so if you installed DNS during setup you probably don't need to be reading this section.

With Windows 2000 already installed, adding the DNS service is accomplished through Control Panel | Add/Remove Programs | Add/Remove Windows Components. When the Windows Components Wizard launches, navigate to Networking Services. Highlight it and click on Details. Select Domain Name System (DNS) and click on OK.

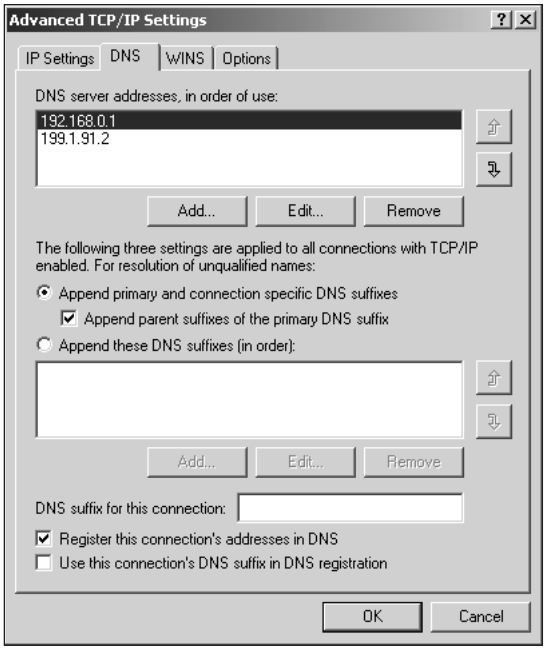

Figure 4-8 You must configure the DNS server IP address in TCP/IP Properties on the server on which you are installing the DNS service

Configuring DNS for Active Directory

With the DNS service installed, the next step is to configure it for use on your Windows 2000 network. You must consider several configuration issues when setting up DNS, such as:

- Root servers
- Forward lookup zones
- Reverse lookup zones
- Resource records
- Dynamic DNS

Root Servers

When you initially launch the DNS Microsoft Management Console (MMC) snap-in, a configuration wizard opens. You first have the option of configuring your server as a root server. As you may recall from earlier in the chapter, root servers on the Internet are authoritative for the entire DNS namespace. Obviously, you would not be able to create a root server that is authoritative for the entire Internet, so you should create a root server only if your network is not connected to the Internet. If your local area network (LAN)

is not connected to the Internet and you create a root server, the root server will be authoritative for any namespace you create.

Forward Lookup Zones

For DNS services to work, at least one forward lookup zone must be configured on your server. Forward lookup zones enable forward lookup queries—the standard method of name resolution in DNS—to work.

To create a forward lookup one, right-click on Forward Lookup Zones in the DNS MMC console and click on New Zone. A configuration wizard will launch. The first choice you have to make when configuring a new zone is what type of zone it will be. The choices are:

- Active Directory Integrated
- Standard Primary
- Standard Secondary

Active Directory Integrated

An Active Directory-integrated zone uses Active Directory to store and manage all zone information. Recall from earlier in the chapter, when we discussed primary and secondary name servers, that a zone can contain only one primary name server, where all updates must be made. This configuration provides a single point of failure, because, if your primary server goes down, you cannot promote a secondary server to become the primary server à la Primary Domain Controllers (PDCs) and Backup Domain Controllers (BDCs) in Windows NT. With an Active Directory-integrated zone, however, all domain controllers (DCs) essentially become primary name servers. Through Active Directory, all DCs are replicated a fully writeable copy of the zone database. This process provides a level of fault tolerance unavailable with a standard primary zone. Also, Windows 2000 clients can register dynamic updates with the nearest available DC, rather than having to contact a single primary server.

Being able to contact any available DC provides additional flexibility. If your network spans several slow wide area network (WAN) links, you would have to set up secondary name servers at the remote locations in order to provide name resolution services with a decent response time. With an Active Directory-integrated zone, your DCs at the remote locations would automatically be able to handle DNS functions.

Standard Primary

A standard primary zone was discussed previously in the chapter—a single primary server that is authoritative for the zone. Whereas in an Active Directory-integrated zone the zone database is stored within the Active Directory, the zone database in a standard primary zone is stored in a text file per RFC standards. If your network will have non-Windows 2000 name servers, you must choose one of the standard zone types

for communication between the Windows 2000 and non-Windows 2000 name servers to take place.

When you go through the configuration wizard to create a standard primary zone, the first step is to create a zone name. Here, you define the namespace for which the zone will be authoritative. After designating the zone name, you are prompted to enter the name of the text file the zone will use. By default, the filename will be *zone*.dns, where *zone* is the zone name you assigned on the previous screen. Notice that you also have the option to use an existing file. If you specify an existing zone database file to use, make sure to copy it to the \winnt\system32\dns directory.

At this point, the zone is configured. You are given the opportunity to review your settings before clicking on Finish.

Standard Secondary

A standard secondary zone was discussed previously in the chapter—it draws its zone information from one or more primary name servers. A secondary zone can contain the zone databases of multiple DNS zones. After specifying the name of the zone in the configuration wizard, you will see the dialog box shown in Figure 4-9.

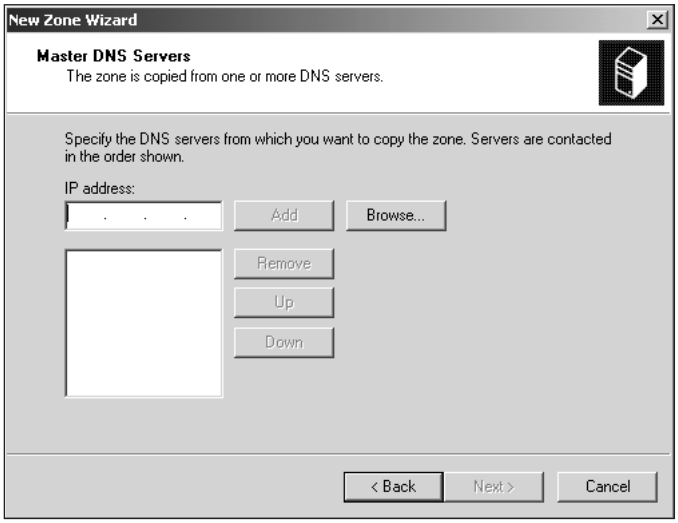

Figure 4-9 When configuring a standard secondary zone, you can specify multiple primary DNS servers from which to draw zone information

After entering the IP addresses of the primary name servers with which this secondary zone should communicate, click on Next. You can review your settings, and then click on Finish.

Reverse Lookup Zones

A reverse lookup zone is not required for DNS services to function; however, you will want to create a reverse lookup zone to allow reverse lookup queries to function. Without a reverse lookup zone, troubleshooting tools such as the **nslookup** command (which can resolve host names from IP addresses) cannot work.

As with forward lookup zones, you have the option of creating Active Directory–integrated, standard primary, or standard secondary zones. No matter which zone type you choose, you name your zone in the window shown in Figure 4-10.

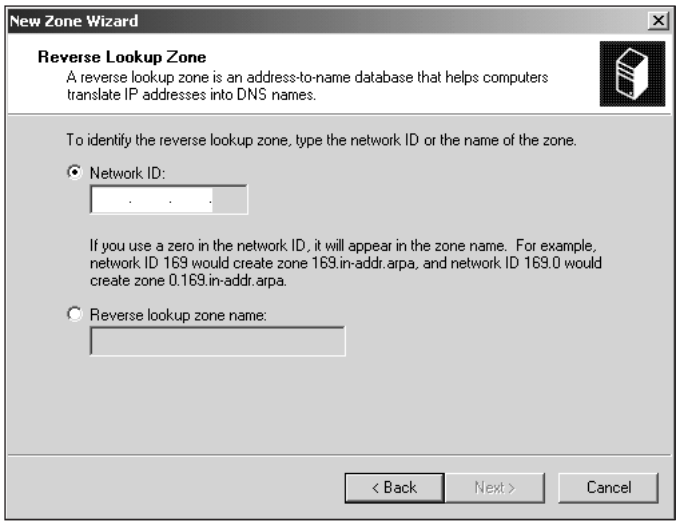

Figure 4-10 Naming a reverse lookup zone to translate IP addresses into host names

Unlike naming a forward lookup zone, you name a reverse lookup zone by its IP address. You can either type your network ID into the first field and watch the reverse lookup zone name automatically be created for you, or you can type the reverse lookup zone name into the second field, following RFC conventions. The on-screen information between Network ID and Reverse Lookup Zone Name describes how to name a reverse lookup zone.

As with forward lookup zones, if you are creating an Active Directory–integrated zone, you are done after supplying the zone name. With standard primary and standard secondary zones, you must also supply the zone filename, which defaults to adding a .dns extension to the end of your zone name. With a standard secondary zone, you must list the IP addresses of the primary DNS servers with which the secondary zone should communicate.

With your zones configured and DNS services functioning, let's look at the entries known as **resource records** that you'll find within the server.

Resource Records

Resource records (RR) are the basic units of information within DNS. When the Windows 2000 DNS service starts up, a number of records are registered at the server. Figure 4-11 shows an example of the netlogon.dns file, which contains a listing of server resource records that are registered with DNS upon startup. The netlogon.dns file is located in the \winnt\system32\config directory.

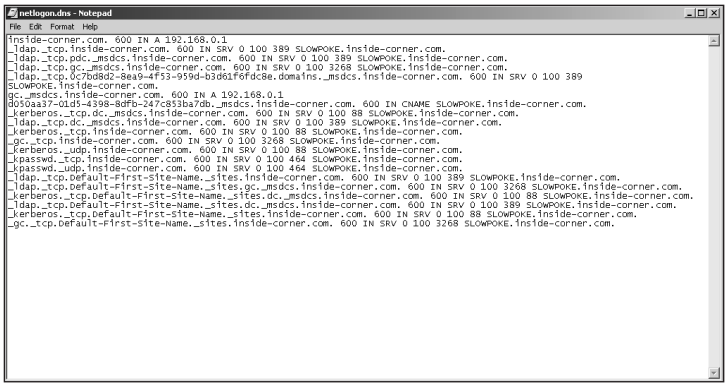

Figure 4-11 A number of resource records are registered with DNS when the service first starts

Okay, so you're probably thinking that the file looks like some foreign language you slept through in high school. The file is actually very structured, so let's look at the first entry:

```
inside-corner.com. 600 IN A 192.168.0.1
```

The structure of the entry's fields is as follows:

- Owner
- TTL
- Class
- Type
- RDATA

Breaking down the first entry in netlogon.dns, **Inside-Corner.com** is the owner. **600** is the TTL, in seconds (10 minutes). **IN** is the class, which is Internet System. You'll find that the class is practically always **IN**. The type identifies the resource record, in this case **A** for address (more on that in a minute). Finally, RDATA is the resource record data. This is a variable whose value depends on the type of record. In an A record such as this, the RDATA entry is the IP address of the system registering the entry.

A number of common resource records are used with Windows 2000 DNS, as follows:

- Start of Authority (SOA)
- Name Server (NS)
- Address (A)
- Pointer (PTR)
- Mail Exchanger (MX)
- Service (SRV)
- Canonical Name (CNAME)

Start of Authority (SOA) Record

The SOA record is contained at the beginning of every zone, both forward lookup and reverse lookup. Its fields define a number of details for the zone, such as:

- *Owner:* Previously defined.
- *TTL:* Previously defined.
- *Class:* Previously defined.
- *Type:* Previously defined.
- *Authoritative Server:* The primary DNS server that is authoritative for the zone.
- *Responsible Person:* The e-mail address of the person who administers the zone.
- *Serial Number:* The serial number of the zone. Remember that the serial number is incremented whenever an update is made, and that secondary servers use serial numbers to determine whether their copy of the zone database is out of date.
- *Refresh:* How often secondary servers check to see if their zone database files need updating.
- *Retry:* How long a secondary server will wait after sending an AXFR or IXFR request before resending the request.
- *Expire:* How long after a zone transfer a secondary server will respond to zone queries before discarding the zone as invalid, due to no communication with the primary server.
- *Minimum TTL:* The minimum TTL a resource record will use if the SOA record does not explicitly state a TTL value.

Name Server (NS) Records

NS records show all servers that are authoritative for a zone: both primary and secondary servers for the zone specified in the SOA record, and primary name servers for any

delegated zones. The NS record uses the Owner, Class, Type, and Authoritative Server fields just described.

Address (A) Records

The A record is the most basic entry in DNS—it maps the FQDN of a host to its IP address. When a client sends a standard forward lookup name resolution query, the server uses A records to resolve the name.

Pointer (PTR) Records

PTR records are the opposite of A records—they provide reverse lookup services for DNS. That is, a PTR record maps an IP address to a FQDN. When a reverse lookup query is sent to a DNS server, such as through the **nslookup** utility, PTR records are consulted to resolve the address.

Mail Exchanger (MX) Records

The MX records designate a mail exchanging server for a DNS zone, which is a host that will process or forward e-mail. In addition to the standard Owner, Class, and Type fields, MX records also support a fourth field: Mail Server Priority. This field is used when your domain has multiple mail servers—mail exchangers with lower values are preferred over mail exchangers with higher values when determining which server to use to process an e-mail message.

Service (SRV) Records

SRV records allow you to specify the location of servers providing a particular service, such as Web servers. You can create SRV records to identify hosts in the zone that provide a service; a resolver can then find the A record of a service to resolve the name. The SRV record format looks like this:

```
_http._tcp.inside-corner.com. IN   SRV 0   0   80
www.inside-corner.com.
```

The fields are as follows:

- *Name:* Designates the name of the service. In this case, **_http.** indicates a Web server.
- *Protocol:* **_tcp.** indicates that TCP protocol is being used. Options are TCP or UDP.
- *Domain:* The domain name to which the resource record refers. In our example, the domain is Inside-Corner.com.
- *Class:* In this case, the value is **IN** (the standard Internet System designation).
- *Type:* The standard Type field. Our example value, **SRV**, indicates an SRV record.

- *Priority:* The first value after the Type (in this case, **0**). As with MX records, hosts with lower priority levels are contacted first.
- *Weight:* The second value after the Type (in this case, **0**). This field is used in conjunction with Priority. When two records have the same priority, the Weight field indicates which server should be tried more frequently. The higher the weight, the more frequently a host will be used.
- *Port:* In this case, **80** is the standard port for the HTTP protocol.
- *Target:* The last field, which is the FQDN of the Web server (in this case, www.inside-corner.com).

Canonical Name (CNAME) Records

A CNAME record creates an alias for a specified host. This type of record is used most commonly to hide implementation details of your network. For example, suppose you have a Web server running at www.mycorp.com. The Web site might really be running on the server server1.mycorp.com. You don't want users to have to use the real server name; and you want the flexibility of being able to move the Web site to a newer, faster server in the future as traffic grows, without having to change the address of your Web site from server1.mycorp.com to server2.mycorp.com. CNAME provides the ability to alias the host name so such problems do not occur.

Other Record Types

There are many other, less common record types besides those listed here. Some are listed in RFCs and some, like WINS and WINS-R, are specific to Windows 2000. In most cases, you will not need to use resource records other than the common ones.

Dynamic DNS

For all of the improvements DNS added over HOSTS files, in essence the update process didn't change. Whenever a new host needed to be added to the network, an administrator had to manually add the appropriate resource records to the primary name server. Fortunately, a relatively recent RFC (RFC 2136) defined a means of dynamically updating the primary server. This RFC relieved a tremendous burden from DNS administration—with the caveat that it works only with Windows 2000 clients. Legacy clients, such as Windows NT 4 and Windows 9*x*, do not support dynamic updates. If you have systems running those operating systems, or other non–Windows 2000 operating systems on your network, you must still add the resource records for those hosts manually.

Dynamic DNS in Windows 2000 is used in conjunction with DHCP. When a Windows 2000 client boots up and receives addressing information from DHCP, it can register itself with DNS, automatically adding the requisite resource records. DDNS is enabled and disabled through zone properties. Simply right-click on your forward or reverse lookup zone and click on Properties. On the General property sheet are the Allow

Dynamic Updates? choices: Yes, No, and Only Secure Updates, as shown in Figure 4-12. When DDNS is enabled, DHCP manages the resource records for DHCP clients. When a DHCP lease expires, the DHCP service cleans up the A and PTR records from DNS.

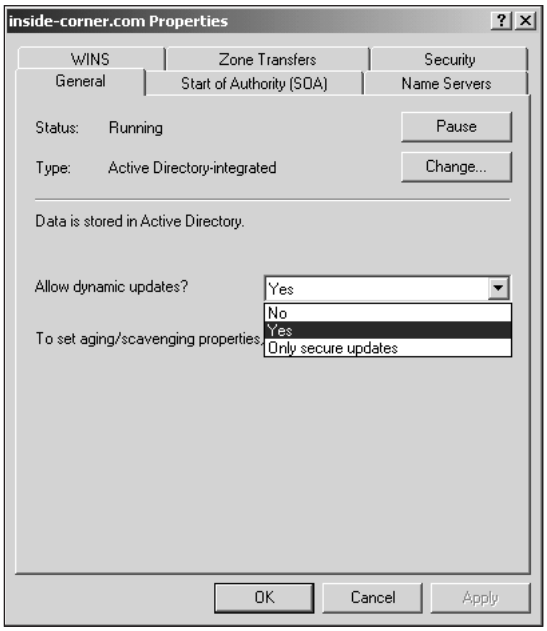

Figure 4-12 Windows 2000 DDNS is configured on a per-zone basis, through zone properties

Monitoring and Troubleshooting DNS for Active Directory

As with any network service, eventually there will be some sort of problem. Windows 2000 provides some tools to monitor and troubleshoot DNS, however, so you can take steps when the DNS service is not behaving as expected.

In Figure 4-13, you can see the monitoring options in Windows 2000. To reach this window, right-click on the DNS server you want to monitor in the DNS MMC snap-in, and then click on Properties. Once there, click on the Monitoring tab. The options and their descriptions are listed in Table 4-2.

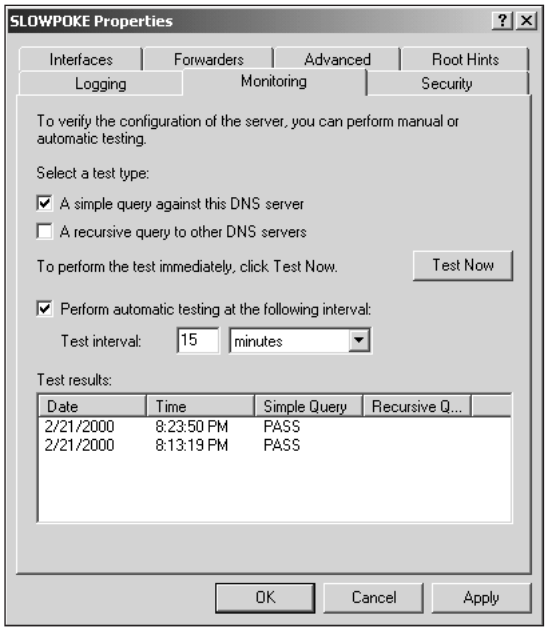

Figure 4-13 Windows 2000 enables an administrator to monitor the DNS service

Table 4-2 Windows 2000 monitoring options and descriptions

Option	Description
Simple Query	As the name implies, a simple forward lookup query is passed to the server for resolution. The result is either PASS or FAIL (as shown in Figure 4-13), and it is time and date stamped in the Test Results box at the bottom of the window.
Recursive Query	This option runs a more complex query, where the server queries other servers until it can resolve the query or runs out of options and fails. With a recursive query, the name server cannot simply refer the client query to another name server.
Perform Automatic Testing	This option tells the server to run the tests you choose at the interval you specify. Doing so is helpful in troubleshooting intermittent server problems.

DNS Logging

In addition to monitoring, you can also enable logging of selected DNS events. You configure logging through the Logging property sheet in the DNS server's properties. In fact, the Logging tab is right next to the Monitoring tab previously discussed. Logging should be performed only for debugging or troubleshooting purposes, because the act

of logging will have a negative impact on server performance and hard disk space. The Logging property sheet is shown in Figure 4-14.

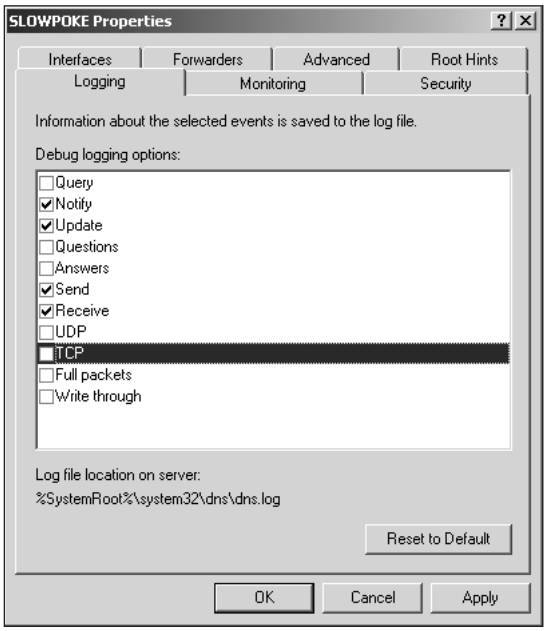

Figure 4-14 Logging is a useful troubleshooting tool when a DNS server is not responding as expected

nslookup

The primary command-line tool for troubleshooting DNS, **nslookup** also makes a handy security tool for tracing hackers back to their source. This basic TCP/IP tool is already on your system if you have the TCP/IP protocol installed.

Remember that if you want nslookup to be able to resolve host names from IP addresses, you must have already configured a reverse lookup zone. nslookup has two modes: noninteractive and interactive. In noninteractive mode, you simply enter a command such as

```
C:\> nslookup 192.168.0.1
```

If nslookup is successful, it will return the host name associated with the IP address in question. There are a number of options for nslookup in noninteractive mode; you can access them by typing "nslookup /?" at a command prompt. One of the more common options is **-server**, which allows you to specify a name server to test other than the current primary DNS server configured on the client.

You enter interactive mode by typing "nslookup" and pressing Enter. To leave interactive mode, type "exit" at a prompt. Typically, you'll use interactive mode when you want more than a single piece of information returned, or when you are running multiple queries one after another.

CHAPTER SUMMARY

DNS is an essential ingredient in a Windows 2000 Active Directory environment. You need to know a lot of theory about how DNS works before you can confidently sit down and configure a DNS server on your network. With the information in this chapter, you should now feel more comfortable with the name resolution process, configuring zones, managing zone transfer for Active Directory–integrated and standard zones, and monitoring and troubleshooting the DNS service if it doesn't function correctly.

Key points to remember from this chapter are as follows:

- DNS is a hierarchical namespace that replaces the flat namespace provided by HOSTS files.
- DNS is a distributed database, meaning a single name server doesn't have to hold the entire database of host-to-IP address mappings.
- The most basic name resolution query is a forward lookup query.
- Reverse lookup queries resolve host names from IP addresses.
- Caching-only servers are not authoritative for any zone and do not hold a zone database file.
- DNS forwarders are used so that only one name server in a zone communicates with DNS servers outside of the local zone.
- A secondary server can be updated by a full zone transfer, incremental transfer, or DNS Notify.
- At least one forward lookup zone must be configured for DNS services to work.
- Resource records are the basic units of information stored in DNS.
- Dynamic DNS requires Windows 2000 clients using DHCP.
- **nslookup** is the primary troubleshooting tool for the DNS service.